F번 - Balanced Bracket Sequence 22점 서브태스크

영어

시간 제한	메모리 제한
1 초 (하단 참고)	256 MB

문제

A "bracket sequence" is a string that consists only of six bracket characters: "(", ")", "[", "]", "{", "}".

"Balanced" bracket sequences are defined as follows:

1. The following three bracket sequences are balanced: "()" "[]" "{}"
2. If two strings "X" and "Y" are balanced bracket sequences, then the strings "XY" and "YX" are also balanced bracket strings.
3. If a string "Z" is a balanced bracket sequence, then the following are also balanced bracket sequences: "(Z)", "{Z}", "[Z]".
4. Lastly, for convenience, an empty string ("") is also defined to be a balanced bracket sequence.

For instance, "[](){}", "[[()]]", and "{}[()]" are balanced bracket sequences, whereas "[][[", "()(}", and "{]" are not.

In this problem, we will consider the strings that contain bracket characters and a wildcard character ("*"). Each wildcard character can be replaced by one of the six bracket characters. For instance, consider the string S = "[**)". If we replace the first wildcard by "]" and the second by "(", then we would obtain a new string S' = "[]()" which is a balanced bracket sequence. If the first wild card is replaced by "(" and the second by "]", then we obtain S'' = "[(])" which is not a balanced bracket sequence.

Given a string that only consists of the six bracket characters and wildcards, you want to turn it into a (longest possible) balanced bracket sequence. You may need to remove some characters and/or replace each wildcard by one of the six bracket characters. For instance, suppose an input string is A = "***". A balanced bracket sequence cannot contain an odd number of characters, so we must remove one of the three characters, and the remaining two wildcards must be replaced by "()" or "[]" or "{}". In another example, consider B = "[{([" where the longest possible balanced bracket sequence would be an empty string (which we can obtain after removing all characters). Lastly, consider C = "(]{})({([)}" where we can remove "]" and "[" to obtain a balanced bracket sequence of length 6 ("({}){()}").

Given a string that consists of bracket characters and wildcards, compute the minimum number of characters you must remove in order to obtain a (longest possible) balanced bracket sequence.

입력

The first line will contain the number of test cases, T.

Each of the following T lines will contain a string.

출력

For each test case, output the minimum number of characters you must remove in order to obtain a balanced bracket sequence.

제한

- 1 ≤ T ≤ 10
- 1 ≤ Length of an input string ≤ 500

서브태스크 1 (6점)

- Each input string will only contain one type of bracket characters and no wildcard characters. That is, each input string will entirely consist of "[]"s or of "{}"s or "()"s with no wildcards.

서브태스크 2 (16점)

- Each input string will only contain the six bracket characters and wildcard character.

예제 입력 1 복사

```
6
***
[{([
(]{})){([)}
**(][)
(*]*[)
[]
```

예제 출력 1 복사

```
1
4
2
2
2
0
```

Cases 1, 2, and 3 are discussed in the problem statement.

Case 4:

The wildcards can be replaced by a pair of brackets (e.g., "{}"), and the square bracket pair "][" can be removed. This gives us a balanced bracket sequence of length 4.

Case 5:

Among many solutions, one is to remove the two square brackets, and to obtain "()()". Another is to replace the first wild card by "[", and remove the second wild card and the subsequent character ("[") to obtain "([])".

Case 6:

If the input string is already balanced, then we do not have to remove any characters.

시간 제한 안내

아래 적혀있지 않은 시간 제한은 언어 도움말 (/help/language)에 적혀있는 기준을 따른다.

- Java: 1초
- Python 3: 4초
- PyPy3: 4초

채점

- 예제는 채점하지 않는다.